

Data247 API Version 3.0

User's Manual

Data247 has a single REST-based API which works for all of our data services. It's easy to use, and yet very powerful, allowing you to access multiple Data247 services in a single API call.

TABLE OF CONTENTS

API REQUEST FORMAT	3
API RESPONSE FORMATS	4
JSON RESPONSES	4
XML RESPONSES	4
CSV RESPONSES	5
YAML RESPONSES	5
TOML RESPONSES	6
MSGPACK RESPONSES	6
DATA247 DATA SERVICES	7
Append247 (Alternate Credit Score)	7
Append247 (Email)	7
Append247 (Gender)	8
Append247 (Name)	8
Append247 (Phone)	9
Append247 (Profile Data)	9
Append247 (Reverse Email)	10
Append247 (Reverse Phone)	10
Append247 (Reverse Zipcode)	11
Append247 (Zipcode)	12
Balance Inquiry	13
Carrier247 (USA) and Text@	13
Carrier247 (International)	14
Carrier247 (Carrier Type)	14
Do-Not-Call (Add)	15
Do-Not-Call (Check)	15
Do-Not-Call (Remove)	15
Locate247 (IP Address)	16
Verify247 (Email)	16

Verify247 (Phone)	17
Verify247 (Postal Address)	17
MULTIPLE SERVICES IN A SINGLE API CALL	19
MULTIPLE QUERIES IN A SINGLE API CALL	19
MULTIPLE PHONE NUMBER PARAMETERS	19
POSTING A JSON DOCUMENT	20
API RESPONSE FORMAT FOR MULTIPLE QUERIES	21
PROFILE DATA FIELDS	23
GENERAL DATA	23
ADDITIONAL CONTACT INFO	23
PROPERTY ATTRIBUTES	23
AUTO ATTRIBUTES	24
PHONE ATTRIBUTES	25
IP ATTRIBUTES	25
DEMOGRAPHIC ATTRIBUTES	25
API ERROR MESSAGES	27
EXAMPLES IN VARIOUS PROGRAMMING LANGUAGES	29
cURL	29
Perl	29
PHP	29
PYTHON	30
JAVA	30
MIGRATING FROM DATA247 API VERSION 2.0 to 3.0	32

API REQUEST FORMAT

Here are some sample API requests (Available via HTTPS GET or POST):

- `https://api.data247.com/v3.0?key=YOURAPIKEY&api=MT&phone=6171234567&addfields=last_port_date`
- `https://api.data247.com/v3.0?key=YOURAPIKEY&api=AP&fname=Joe&lname=Smith&address=24 Main Street&City=Boston&state=MA&zip=02134&out=yaml`
- `https://api.data247.com/v3.0?key=YOURAPIKEY&api=CU,AN,APR&phone=16172223333&out=xml`

There are 6 main parts to the API queries:

- **The URL:** This remains the same for each query: `https://api.data247.com/v3.0?`
- **The API key:** Each API user has their own unique API key(s) which identifies them. If you don't know your API key, log-into your Data247 account, go to the "Account" menu, and select "API Key / Password". If you don't see an API key in the list on this page, you can generate a new one by entering a description and then pressing the "Generate API Key" button.
- **The API Service Code(s):** This identifies the particular API data service or services you wish to use. Data247 offers many different data services such as Text@, phone append, IP Geolocation, etc. For the full list, see our "Data247 Data Services" section below.
- **The Input Data:** For each Data247 service, you provide input data and we reply with a response based on your input. The input parameters will vary for each service. See the "Data247 Data Services" section below for more information.
- **The Field Specifiers:** There are two types of field specifiers. "fields=" and "addfields=". The former specifies ALL data fields to be returned by the query, and the latter specifies which optional fields should be returned alongside the standard return fields. In both cases, the fields should be separated by commas. Field specifiers are optional. If omitted, only the standard data fields will be returned.
- **The Response Format:** Either xml, json, yaml, toml, msgpack, or csv. See the "API RESPONSE FORMATS" section for more details. The response format specifier is optional. If omitted, JSON format will be returned by default.

Note: To do multiple queries in a single API call, see the "MULTIPLE QUERIES IN A SINGLE API CALL" section below.

API RESPONSE FORMATS

The API can currently support 6 different output formats: JSON, XML, YAML, TOML, MsgPack, and CSV. JSON is the default response format.

You can specify the API response format using the “out=” parameter in the API request, as follows (note they are all lower-case):

- out=json
- out=xml
- out=yaml
- out=toml
- out=csv
- out=msgpack

JSON RESPONSES

A JSON response encapsulates the entire response in a “response” tag. Inside the response tag will always be a “status” indicator. If the status is anything other than “OK”, the query failed. See the “API ERROR MESSAGES” section below for more information on errors. If the status is “OK”, then the results from the query will appear as an array of “results”, as shown below:

```
{
  "response": {
    "status": "OK",
    "results": [
      {
        "phone": "15085551212",
        "wless": "n",
        "carrier_name": "Neutral Tandem-Illinois, LLC - IL",
        "carrier_id": 50724,
        "sms_address": "",
        "mms_address": ""
      }
    ]
  }
}
```

XML RESPONSES

An XML response encapsulates the entire response in a “<response>” tag. Inside the response tag will always be a “status” indicator. If the status is anything other than “OK”, the query failed. See the “API ERROR MESSAGES” section below for more information on errors. If the status is

“OK”, then the results from the query will appear as an array of “<results>”, each encapsulated in a “<result>” tag, as shown below:

```
<response>
  <status>OK</status>
  <results>
    <result>
      <phone>1505551212</phone>
      <wless>n</wless>
      <carrier_name>Neutral Tandem-Illinois, LLC - IL</carrier_name>
      <carrier_id>50724</carrier_id>
      <sms_address/>
      <mms_address/>
    </result>
  </results>
</response>
```

CSV RESPONSES

A CSV response is similar to a .csv (comma separated values) file format. It contains a header row with each column name, in quotes, separated by commas. Similarly, each result will appear in its own row, with each data element in quotes, separated by commas.

```
"phone","wless","carrier_name","carrier_id","sms_address","mms_address"
"15084041018","n","Neutral Tandem-Illinois, LLC - IL","50724","",""
```

YAML RESPONSES

A YAML response contains the entire response under the “response” tag. Inside the response tag will always be a “status” element. If the status is anything other than “OK”, the query failed. See the “API ERROR MESSAGES” section below for more information on errors. If the status is “OK”, then the results from the query will appear as an array of ‘results’, as shown below:

```
response:
  results:
  - carrier_id: 40654
    carrier_name: Sprint Spectrum, L.P.
    mms_address: 7818641234@tmomail.net
    phone: '17818641234'
    sms_address: 7818641234@tmomail.net
    wless: y
  status: OK
```

TOML RESPONSES

A TOML response contains the entire response under the “response” tag. Inside the response tag will always be a “status” element. If the status is anything other than “OK”, the query failed. See the “API ERROR MESSAGES” section below for more information on errors. If the status is “OK”, then the results from the query will appear as an array of ‘results’, as shown below:

```
[response]
status = "OK"
[[response.results]]
phone = "17818641234"
wless = "y"
carrier_name = "Sprint Spectrum, L.P."
carrier_id = 40654
sms_address = "7818641234@tmomail.net"
mms_address = "7818641234@tmomail.net"
```

MSGPACK RESPONSES

MsgPack (<https://msgpack.org/>) uses binary characters as field separators, and tends to be more compact than other formats. The MsgPack response contains the entire response under the “response” tag. Inside the response tag will always be a “status” element. If the status is anything other than “OK”, the query failed. See the “API ERROR MESSAGES” section below for more information on errors. If the status is “OK”, then the results from the query will appear as an array of ‘results’, as shown below:

```
\x81\xa8response\x82\xa6status\xa2OK\xa7results\x91\x86\xa5phone\xab17818641234\xa5wless\xa1y\xaccarrier_name\xbeBellsouth Mobility, LLC
(AT+T)\xaacarrier_id\x06\xabsms_address\xb67818641234@txt.att.net\xabmms_address\xb67818641234@mms.att.net
```

DATA247 DATA SERVICES

The various Data247 services are listed below, alphabetically:

Append247 (Alternate Credit Score)

Description: Returns an “alternate” credit score for a consumer, which does not use data from the “big 3” credit bureaus.

Service Code: **AC**

Input Data Fields:

- fname: Consumer’s first name
- lname: Consumer’s last name
- address: Consumer’s street address
- city: Consumer’s city
- state: Consumer’s state
- zip: Consumer’s zip code (optional)
- email: Consumer’s email address (optional)
- phone: Consumer’s phone number (optional)

Standard Return Fields:

- credit_score: An alternate credit score ranging from 430 - 830
- profile_data: See “PROFILE DATA FIELDS” section below

Optional Return Fields:

- billable - Indicates whether customer was charged for this query
- cost - Cost of this query

Append247 (Email)

Description: Returns the email address associated with a name and address (business or residential)

Service Code: **AE**

Input Data Fields:

- fname: Person’s first name
- lname: Person’s last name
- name: Full name; may be used instead of fname and lname
- business_name - If business address (optional)
- address: Contact’s street address
- city: Contact’s city
- state: Contact’s state
- zip: Contact’s zip code (optional)
- phone: Contact’s phone number (optional)

Standard Return Fields:

- firstname: Contact's first name
- lastname: Contact's last name
- address: Contact's address
- city: Contact's city
- state: Contact's state
- zip: Contact's zip code
- email: Contact's email address

Optional Return Fields:

- billable - Indicates whether customer was charged for this query
- cost - Cost of this query
- website: Company website (business listings only)
- num_employees: Total employees (business listings only)
- total_revenue: Total revenue (business listings only)
- title: Contact's business title (business listings only)
- business_type: Type of business (business listings only)

Append247 (Gender)

Description: Returns a person's probable gender based on their first (given) name

Service Code: **AG**

Input Data Fields:

- fname - Person's first name
- name - Person's full name

Standard Return Fields:

- fname - Person's first name
- gender - Probable gender (MALE or FEMALE)
- gender_pct - The percent likelihood of gender being correct.

Optional Return Fields:

- billable - Indicates whether customer was charged for this query
- cost - Cost of this query

Append247 (Name)

Description: Returns the name associated with a phone number

Service Code: **AN**

Input Data Fields:

- phone: phone number

Standard Return Fields:

- name - Person's first name

Optional Return Fields:

- billable - Indicates whether customer was charged for this query

- cost - Cost of this query

Append247 (Phone)

Description: Returns the phone number(s) associated with a name and address
(business or residential)

Service Code: AP

Input Data Fields:

- fname: Person's first name
- lname: Person's last name
- name: Full name; may be used instead of fname and lname
- business_name - If business address (optional)
- address: Contact's street address
- city: Contact's city
- state: Contact's state
- zip: Contact's zip code (optional)
- email: Contact's email address (optional)

Standard Return Fields:

- firstname: Contact's first name
- lastname: Contact's last name
- address: Contact's address
- city: Contact's city
- state: Contact's state
- zip: Contact's zip code
- phone: Contact's phone number
- type: Type of phone (Landline, Wireless, or VOIP)
- addtl: An array of up to five additional phone numbers with phone type.

Optional Return Fields:

- billable - Indicates whether customer was charged for this query
- cost - Cost of this query
- website: Company website (business listings only)
- num_employees: Total employees (business listings only)
- total_revenue: Total revenue (business listings only)
- title: Contact's business title (business listings only)
- business_type: Type of business (business listings only)

Append247 (Profile Data)

Description: Returns profile data for a consumer

Service Code: AF

Input Data Fields:

- fname: Consumer's first name

- lname: Consumer's last name
- address: Consumer's street address
- city: Consumer's city
- state: Consumer's state
- zip: Consumer's zip code (optional)
- email: Consumer's email address (optional)
- phone: Consumer's phone number (optional)

Standard Return Fields:

- See "PROFILE DATA FIELDS" section below for a complete list of return fields.

Optional Return Fields:

- billable - Indicates whether customer was charged for this query
- cost - Cost of this query

Append247 (Reverse Email)

Description: Returns the name and address associated with an email address.

Service Code: **AER**

Input Data Fields:

- email: Contact's email address

Standard Return Fields:

- phone - Contact's phone number
- email - Contact's email address
- firstname - Contact's first name
- lastname - Contact's last name
- m_initial - Contact's middle initial
- business - Contact's business name
- address - Contact's address
- city - Contact's city
- state - Contact's state
- zip - Contact's zip code
- zip4 - Additional 4 digits of contact's zip code
- addrtype - business or residential
- gender - MALE or FEMALE
- gender_prct - Percent likelihood of gender being correct

Optional Return Fields:

- billable - Indicates whether customer was charged for this query
- cost - Cost of this query

Append247 (Reverse Phone)

Description: Returns the name and address associated with a phone number.

Service Code: **APR**

Input Data Fields:

- phone: Contact's phone number

Standard Return Fields:

- phone - Contact's phone number
- email - Contact's email address
- firstname - Contact's first name
- lastname - Contact's last name
- m_initial - Contact's middle initial
- business - Contact's business name
- address - Contact's address
- city - Contact's city
- state - Contact's state
- zip - Contact's zip code
- zip4 - Additional 4 digits of contact's zip code
- addrtype - business or residential
- gender - MALE or FEMALE
- gender_prct - Percent likelihood of gender being correct

Optional Return Fields:

- billable - Indicates whether customer was charged for this query
- cost - Cost of this query

Append247 (Reverse Zipcode)

Description: Returns various information about a zipcode

Service Code: **AZR**

Input Data Fields:

- zip - Either a 5 or 9 digit zipcode (NNNNN or NNNNN-NNNN)

Standard Return Fields:

- zip - Zip code
- zip4 - Last 4 digits of zipcode
- streetnum - Street number or number range of address
- predir - Direction indicator before street name
- street - The street name of the address
- street_suffix - Like St, Rd, Pl, etc.
- postdir - Direction indicator after street name
- unit - Unit number or range of street address
- city - Name of city
- state - Name of state
- county - Name of county
- PRUrbCode" - Puerto Rico Urbanization Code
- country - Country of zipcode (Currently service only works for USA)

- FIPS - Fips code
- company_name - If zipcode belongs to a company
- congress_dist - Congressional district served by zip code
- areacode - areacode served by zip code
- timezone - Timezone observed in zip code
- DPBC - DPBC code

Optional Return Fields:

- billable - Indicates whether customer was charged for this query
- cost - Cost of this query

Append247 (Zipcode)

Description: Returns the zipcode for an address

Service Code: AZ

Input Data Fields:

- address - Street address
- city - City
- state - State

Standard Return Fields:

- streetnum - Street number or number range of address
- predir - Direction indicator before street name
- street - The street name of the address
- street_suffix - Like St, Rd, Pl, etc.
- postdir - Direction indicator after street name
- unit - Unit number or range of street address
- city - Name of city
- state - Name of state
- zip - Zip code
- zip4 - Additional 4 zip code digits
- county - Name of county
- PRUrbCode" - Puerto Rico Urbanization Code
- country - Country of zip code (Currently service only works for USA)
- FIPS - Fips code
- company_name - If zip code belongs to a company
- congress_dist - Congressional district served by zip code
- areacode - areacode served by zip code
- timezone - Timezone observed in zip code
- DPBC - DPBC code

Optional Return Fields:

- billable - Indicates whether customer was charged for this query
- cost - Cost of this query

Balance Inquiry

Description: Returns your current account balance

Service Code: **B**

Input Data Fields: None

Standard Return Fields:

- balance - Current account balance

Optional Return Fields:

- billable - Indicates whether customer was charged for this query
- cost - Cost of this query

Carrier247 (USA) and Text@

Description: Carrier Lookup for USA/Canada with email-to-sms gateway addresses.

Service Code: **CU** or **MT**

Input Data Fields:

- phone - The phone number to query

Standard Return Fields:

- phone - The phone number queried
- wless - Either "y" for wireless or "n" for non-wireless
- carrier_name: carrier name of provided number
- carrier_id:: Carrier id of carrier
- sms_address: SMS gateway address of number
- mms_address: MMS gateway address of number

Optional Return Fields:

- billable - Indicates whether customer was charged for this query
- cost - Cost of this query
- type - Either "M" for mobile phone, "L" for landline, or "V" for VOIP
- GIT - Voicemail greeting interrupt Tone
- ocn - The operating company number
- timezone - The timezone in the area specified by the phone number's NPA
- tz_offset - The offset from UTC time in the area specified by the numbers NPA
- iana - IANA timezone of the area specified by the number's NPA
- last_port_date - The date the phone was last ported to a new carrier
- ratecenter - BROKEN
- 'clli' - BROKEN
- assign_date - BROKEN,
- prefix_type - BROKEN
- switch_name - BROKEN,
- switch_type - BROKEN
- lata - BROKEN
- dba - Standard name for carrier
- mno - Mobile Network Operator

- city - City where NPA/NXX is located
- state - State where NPA/NXX is located
- county - Country where NPA/NXX is located
- zip - Zip code where NPA/NXX is located

Carrier247 (International)

Description: Worldwide carrier lookup service

Service Code: **CI**

Input Data Fields:

- phone - The phone number to query (must include country code)

Standard Return Fields:

- phone - The phone number queried
- wless - Either “y” for wireless or “n” for non-wireless
- carrier_name: carrier name of provided number
- carrier_id:: Carrier id of carrier
- country - The country of the phone number
- isoCountry2 - The ISO 3166 2-digit country code:
https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2

Optional Return Fields

- billable - Indicates whether customer was charged for this query
- cost - Cost of this query
- type - Either “M” for mobile phone, “L” for landline, or “V” for VOIP
- GIT - Voicemail greeting interrupt Tone
- sms_address - The email-to-sms gateway address
- mms_address - The email-to-mms gateway address

Carrier247 (Carrier Type)

Description: Returns whether a phone number is wireless, landline or VOIP

Service Code: **CT**

Input Data Fields:

- phone - The phone number to query (must include country code)

Standard Return Fields:

- phone - The phone number queried
- type - Either “M” for mobile phone, “L” for landline, or “V” for VOIP

Optional Return Fields:

- billable - Indicates whether customer was charged for this query
- cost - Cost of this query

Do-Not-Call (Add)

Description: Adds a phone number to customer's internal do-not-call list.

Service Code: **DA**

Input Data Fields:

- phone - Phone number to be added to customer's internal DNC list

Standard Return Fields:

- phone - Phone number which was added
- action - Will have a value of "ADDED" on success

Optional Return Fields:

- billable - Indicates whether customer was charged for this query
- cost - Cost of this query

Do-Not-Call (Check)

Description: Checks if phone appears on Federal or customer's internal DNC List.

Service Code: **DC**

Standard Return Fields:

- phone - Phone number which was checked
- dnc - Will be one of the following values:
 - OK-TO-DIAL - Phone doesn't appear on Federal or internal list
 - FEDERAL DNC - Phone appears on Federal list
 - INTERNAL DNC - Phone appears on customer's internal DNC list
 - UNAUTHORIZED - Customer doesn't have access to this areacode in the Federal DNC list.

Optional Return Fields:

- billable - Indicates whether customer was charged for this query
- cost - Cost of this query

Do-Not-Call (Remove)

Description: Removes a phone number from customer's internal do-not-call list.

Service Code: **DR**

Input Data Fields:

- phone - Phone number to be removed from customer's internal DNC list

Standard Return Fields:

- phone - Phone number which was removed
- action - Will have a value of "REMOVED" on success

Optional Return Fields:

- billable - Indicates whether customer was charged for this query
- cost - Cost of this query

Locate247 (IP Address)

Description: Finds the geographical location of an IP address

Service Code: **GI**

Input Data Fields:

- ip - The IP address being queried

Standard Return Fields:

- ip_address - The IP address being queried
- city - The city where the IP address is located
- state - The state where the IP address is located
- country - The country where the IP address is located
- zipcode - The zipcode where the IP address is located
- latitude - The latitude of the IP address
- longitude - The longitude of the IP address
- areacode - The areacode where the IP address is located (if in USA)
- timezone - The timezone where the IP address is located
- dst - Whether DST is active where the IP address is located

Optional Return Fields

- billable - Indicates whether customer was charged for this query
- cost - Cost of this query

Verify247 (Email)

Description: Checks if an email address is valid and deliverable

Service Code: **VE**

Input Data Fields:

- email - The email address to be verified

Standard Return Fields:

- email - The email address which was verified
- valid - YES or NO - whether it's a valid, sendable email address
- reason - Description why/why not address is valid
- free_account - YES/NO - Whether it's a free email account (yahoo, gmail, etc)

Optional Return Fields:

- billable - Indicates whether customer was charged for this query
- cost - Cost of this query
- disposable_email - YES/NO - Whether it's a temporary email address

Verify247 (Phone)

Description: Verifies a phone number is in-service and accepts inbound calls

Service Code: VP

Input Data Fields:

- phone - The phone number to be verified

Standard Return Fields:

- phone - The phone number which was verified
- active - YES/NO - whether phone is active
- confidence - HIGH/LOW - Confidence level

Optional Return Fields

- billable - Indicates whether customer was charged for this query
- cost - Cost of this query

Verify247 (Postal Address)

Description: Checks a USA postal address and returns corrections to make it fit USPS standards.

Service Code: VA

Input Data Fields:

- address - Street address
- city - City name
- state - State
- zip - Zip code (nnnnn or nnnnn-nnnn)

Standard Return Fields:

- valid - YES/NO - Is address valid?
- addr1 - Line 1 of corrected street address
- addr2 - Line 2 of corrected street address
- streetnum - Street number or number range of address
- predir - Direction indicator before street name
- street - The street name of the address
- street_suffix - Like St, Rd, Pl, etc.
- postdir - Direction indicator after street name
- unit - Unit number or range of street address
- city - Name of city
- state - Name of state
- zip - Zip code
- zip4 - Additional 4 zip code digits
- county - Name of county
- PRUrbCode" - Puerto Rico Urbanization Code
- country - Country of zip code (Currently service only works for USA)
- FIPS - Fips code
- company_name - If zip code belongs to a company

- congress_dist - Congressional district served by zip code
- areacode - areacode served by zip code
- timezone - Timezone observed in zip code
- DPBC - DPBC code

MULTIPLE SERVICES IN A SINGLE API CALL

The Data247 API allows multiple data services to be used in a single API call. For example, someone may want to use both Phone and Email append services to get a consumer's phone and their email address. The resulting query would look like this:

```
https://api.data247.com/v3.0?key=YOURAPIKEY&api=AP,AE&fname=Joe&lname=Smith&address=12 Main Street&city=Nashua&state=NH
```

The result for this query would contain all of the return fields from the AP query and all of the return fields from the AE query. Any return fields which are returned by both services will only appear once in the return data.

Here's a more interesting example:

```
https://api.data247.com/v3.0?key=YOURAPIKEY&api=AN,AG&phone=16175551212
```

The AN service requires a phone number and returns a name. The AG service requires a name and returns a gender. But only the phone number is being passed in. The API is smart enough to know that the name required by the AG service can be obtained from the result of the AN service.

MULTIPLE QUERIES IN A SINGLE API CALL

The "API Request Format" section above explains how to send one query at a time using the API, but the API is capable of sending multiple queries at once. Depending on the service type, there are one or two ways to do this:

MULTIPLE PHONE NUMBER PARAMETERS

This method only applies to the following services:

- Text@
- Carrier247 (USA)
- Carrier247 (International)
- Carrier247 (Carrier Type)
- Do-Not-Call
- Append247 (Reverse Phone)

For the above services, one can submit multiple phone numbers by listing them in the URL as p1=<phone>&p2=<phone>&p3=<phone> . . .

Using this method, up to 99 phone numbers may be specified in a single API call. Here's an example:

```
https://api.data247.com/v3.0?key=MyAPIKey&api=DC&p1=15085551212&p2=17815551212&p3=16035551212&p4=16175551212
```

POSTING A JSON DOCUMENT

This second method works with all API services. You may call the API as an HTTPS POST, and instead of adding the input data as URL parameters, they may instead be POSTed as a JSON-format document, as follows:

```
{
  "XX": [
    {
      "query 1 parameter": "value",
      "query 1 parameter": "value",
      "query 1 parameter": "value",
      "query 1 parameter": "value"
    },
    {
      "query 2 parameter": "value",
      "query 2 parameter": "value",
      "query 2 parameter": "value",
      "query 2 parameter": "value"
    },
    {
      "query 3 parameter": "value",
      "query 3 parameter": "value",
      "query 3 parameter": "value",
      "query 3 parameter": "value"
    }
  ]
}
```

. . . where "XX" is the service code.

The API will allow up to 2,500 queries in a single JSON document, but note that some Data247 services are slower than others, and for some it could cause a timeout to occur.

Here's an example JSON which could be POSTed to the API:

```
{
  "AP": [
    {
      "firstname": "Joe",
      "lastname": "Smith",
      "address": "12 Main Street",
      "city": "Dallas",
      "state": "TX",
      "zip": "75201"
    },
    {
      "firstname": "Sarah",
      "lastname": "Jones",
      "address": "10 Elm Street",
      "city": "Fitchburg",
      "state": "MA",
      "zip": "01420"
    }
  ]
}
```

API RESPONSE FORMAT FOR MULTIPLE QUERIES

When multiple queries are sent in a single API call, the API response will differ in two ways:

1. The array of “results” in the response will contain multiple items instead of just one.
2. In addition to the “status” indicator at the response level, there will also be a “status” indicator present in the result set for each query. This query-level status will indicate the status of that particular query. A status of “OK” means the query was successful. Otherwise the status will contain an error message (see API ERROR MESSAGES section below)

Here's an example of a response for multiple queries in JSON format:

```
{
  response :{
    status : "OK",
    results :[
      {
        status : "OK",
        phone : "15084041234",
        wless : "n",
        carrier_name : "Bandwidth.com",
        carrier_id : 50025,
        sms_address : "",
        mms_address : ""
      },
      {
        status : "OK",
        phone : "17818640001",
        wless : "y",
        carrier_name : "Bellsouth Mobility, LLC (AT+T)",
        carrier_id : 6,
        sms_address : "7818640001@txt.att.net",
        mms_address : "7818640001@mms.att.net"
      }
    ]
  }
}
```

PROFILE DATA FIELDS

GENERAL DATA

FName - First Name [20 chars]
LName - Last Name [20 chars]
MName - Middle Initial [1 Char]
BusName - Business Name [100 chars]
House - Primary House Number [10 chars]
PreDir - Street Pre Direction [2 chars]
Street - Street name [28 chars]
StrType - Street suffix: ST, AVE,etc [4 chars]
PostDir -Street Post Direction [2 chars]
AptType -Secondary Unit designator [4 chars]
AptNbr -Secondary unit number [8 chars]
City - USPS City Name [28 chars]
State -USPS state abbreviation [2 chars]
Zip -numeric USPS zip code [5 chars]
Z4 -numeric USPS zip+4 [4 chars]
DPC -Delivery point code with check digit [3 chars]
CRTE -Carrier Route [4 chars]
CNTY - FIPS county code [3 chars]
Z4Type -USPS Zip+4 type: [1 char]
 F – firm or company address
 G – General delivery address
 H – High-rise or business complex
 P – PO Box address
 R – Rural route address
 S – Street or residential address
 Blank - Unknown
DPV - Delivery Point Validation
 Y – Address DPV confirmed for both
 primary and (if present) secondary numbers
 D – Address DPV confirmed for primary number only,
 secondary number information was missing
 S – Address DPV confirmed for the primary number
 only, and secondary number information was present
 but unconfirmed
 N – Both Primary and (if present) Secondary number
 information failed to DPV confirm
 Blank – Address not presented to hash table
 Deliverable Deliverable flag [y, n, or blank]
ValDate - Last addr. validation date [YYYYMMDD]

ADDITIONAL CONTACT INFO

Phone - Phone (up to 3 additional Phones) [10 chars]
PhoneType - Phone Type (up to 3, one for each phone)
 L – LandLine,
 V - VoIP,
 W – Wireless, O -
 Other
Category -Matched Category (up to 3, one for each phone)
 I – Individual
 H – Household
 A – Address
 Z – Name/Zip

Email - Email (up to 3 additional Emails) [100 chars]
Suppression Email Suppression Code
 D - Files with domains only
 C - Cops
 B - Bounce
 T - Traps
 O - Opt out
 G - General
 S - Complainers
 I - other
 N - None
URL - Indicates the website the consumer "opted-in" to
 receive marketing emails [100 chars]
ODate - Email Last Seen Date [YYYYMMDD]

PROPERTY ATTRIBUTES

PROP_IND - Property type indicator [2 digit numeric]
 10 = Single Family / Townhouse
 11 = Condominium (residential)
 20 = Commercial
 21 = Duplex, Triplex, Quadplex
 22 = Apartment
 23 = Hotel, Motel
 24 = Commercial (condominium)
 25 = Retail
 26 = Services (general public)
 27 = Office Building
 28 = Warehouse
 29 = Financial Institution
 30 = Hospital (medical complex, clinic)
 31 = Parking
 32 = Amusement – Recreation
 50 = Industrial
 51 = Industrial Light>
 52 = Industrial Heavy
 53 = Transport
 54 = Utilities
 70 = Agricultural
 80 = Vacant
 90 = Exempt nn
PROP_VALCALC - The "total" (i.e., land + improvement)
 value closest to current market value used for
 assessment by county or local taxing authorities.
 Integer [dollars]
PROP_IMP_VALCALC - The "improvement" value closest to
 current market value used for assessment by county
 or local taxing authorities. [Integer (dollars)]
PROP_IMP_VALCALC_IND - "improvement" value
 A = Assessed
 M = Market
 P = Appraised
 T = Transitional
PROP_ASSED_VAL - The Total Assessed Value of the

Parcel's Land & Improvement values as provided by the county or local taxing/assessment authority.
Integer [dollars]

PROP_ACRES - Total land mass in acres. (4 decimal points). Example: 13000=1.3 acres. [Integer]
PROP_LANDSQFT - Total landmass in Square Ft [int]
PROP_YRBLD - The construction year of the original building. [YYYY]
PROP_LIVINGSQFT -The area of a building that is used for general living. This is typically the area of a building that is heated or air conditioned and does not include Garage, Porch or Basement square footage. [Integer]
PROP_RMS -Total number of rooms contained in the primary building. [Integer]
PROP_BEDRMS -Total number of bedrooms contained in the primary building. [Integer]
PROP_BATHS - Total number of bathrooms, 2 implied decimal places. 2.00 baths = 200. [Integer]
PROP_FULLBATHS - Total number of Full Baths (typically comprised of a sink, toilet, and bathtub / shower stall). A home containing 2 1/2 baths would have the number 2 stored in this field. [Integer]
PROP_HALFBATHS - Total number of Half Baths (typically comprised of a sink & toilet). A home containing 2 1/2 baths would have the number 1 stored in this field. [Integer]
PROP_AC - The type of air conditioning method used to cool the building.
 0=AC.NONE
 ACA = AC.COMMERCIAL A/C
 ACE = AC.CENTRAL
 ACH = AC.CHILLED WATER
 ACP = AC.CENTRAL PARTIAL
 ACW = AC.CENTRAL & UNIT
 ADU = AC.DUAL UNIT
 AEV = AC.EVAPORATIVE
 AF = AC.FAN COOLING
 AHT = AC.HEAT PUMP
 AOF = AC.OFFICE ONLY
 APF = AC.REFRIGERATION
 APK = AC.PACKAGE
 APR = AC.PACKAGE ROOF
 APT = AC.PARTIAL
 APV = AC.REFRIGERATION / EVAPORATION
 ASE = AC.SEPARATE SYSTEM
 ASO = AC.SOLAR
 ASP = AC.SPLIT SYSTEM
 AWA = AC.WALL UNIT
 AWI = AC.WINDOW UNIT
 AWN = AC.WALL/WINDOW UNIT [3 chars]
PROP_FRPL - This field is populated with a "Y" if a fireplace is located within the building. [Y or blank]
PROP_POOL - Populated with a "Y" if a Pool is present on the parcel. [Y or blank]
PROP_ROOFTYPE - Type of roof
 999 = BYPASS
 9A0 = IRREGULAR
 9B0 = LEAN TO
 A = A-FRAME

B = BARN
 C = CANOPY
 D = DORMER
 E = FRAME
 F = FLAT
 G = GABLE
 H = GABLE/HIP
 I = HIP
 J = GEODESIC
 K = MANSARD
 L = BARREL
 M = MONITOR
 N = CONTEMPORARY
 O = SHED
 P = PITCHED
 Q = PYRAMID
 R = ARCHED
 S = SAWTOOTH
 T = CATHEDRAL / CLERESTORY
 U = BUBBLE
 V = GAMBREL
 W = SWISS CHALET / ALPINE
 X = COMPLEX/CUSTOM
 Y = BUTTERFLY
 Z = GAMBREL / MANSARD (3 chars)

PROP_TAXAMT - The tax amount provided by the county or local taxing / assessment authority [Int (dollars)]

PROP_RECDDATE - The date the sales transaction was record at the county [YYYYMMDD]

PROP_SALEAMT - Price of the sale as depicted on the recorded sales transaction. [Integer (dollars)]

PROP_MTGAMT - Amount of loan. [Integer (dollars)]

PROP_MTGDATE - Date mortgage was initiated. [YYYYMMDD]

PROP_MTGTERM - The length of time of the mortgage in years. [Integer (years)]

PROP_MTGDUEDATE - Date mortgage becomes due. [YYYYMMDD]

AUTO ATTRIBUTES

Make - Vehicle Make, [30 chars max]

Model - Vehicle Model, [30 chars max]

Year - Vehicle Year [YYYY]

ClassCD - Vehicle Class Code, [15 chars max]

FuelTypeCD - Vehicle Fuel Code [1 character]

MFGCD - Vehicle Manufacturing Code [1 character]

StyleCD - Vehicle Style Code [10 chars]

Mileage - Mileage from the last odometer reading in increments of 10,000. [A-Z]

A=0-10,000.

Z= 250,000+.

ODate - Last verification date. [YYYYMMDD]

IN_financing_prob_rank - In Market for new Financing [0-100]

100 - most qualified

10 - least qualified

0 - Unknown

IN_insurance_prob_rank - In Market for getting Vehicle

Insurance [1-100]
 100 - most qualified
 10 - least qualified
 0 - Unknown

IN_new_vehicle_prob_rank - In Mrket for new Vehicle [1-100]
 100 - most qualified
 10 - least qualified
 0 - Unknown

IN_auto_parts_prob_rank - In Market for auto parts [1-100]
 100 - most qualified
 10 - least qualified
 0 - Unknown

IN_used_vehicle_prob_rank - In Mket for used Vehicle [1-100]
 100 - most qualified
 10 - least qualified
 0 - Unknown
 0-100

PHONE ATTRIBUTES

PhoneType - Phone type for the Phone.
 L=Landline
 V=VoIP
 W=Wireless
 O=Other.

Phonetype2 - Phone type for the 2nd Phone.
 L=Landline
 V=VoIP
 W=Wireless
 O=Other

IP ATTRIBUTES

IP - IP v4 address. [xxx.xxx.xxx.xxx]
IPRegion - Region. State name if within the US. [50 chars]
City - Approximate city the IP address is located in [28 chars]
Latitude - Approximate latitude of the IP address. [11 chars]
Longitude - Approximate longitude of the IP address. [11 chars]
Zip - Approx. Zip code associated with the IP address [5 chars]
ISP - Internet service provider for IP address [255 chars]
IPFrom - Starting IP of the range that the IP address belongs
 [IP4 address: xxx.xxx.xxx.xxx]
IPTo - Final IP of the range that the IP address belongs
 [IP 4 address: xxx.xxx.xxx.xxx]

DEMOGRAPHIC ATTRIBUTES

Gender - M=male, F=Female, blank or U=unknown. [1 char]
Age - Adult Age [0 - 99]
LOR - Length of Residence in years, [2 digits]
Homeowner - Homeowner status [1 char]
 H=Homeowner
 R=Renter
 A = Inferred Homeowner
 B = Inferred Renter
MedYrBld - The year home was built. [4 digits]
EHI - Estimated household income, [1 char]

A = Less than \$20,000
 B = \$20,000 - \$29,999
 C = \$30,000 - \$39,999
 D = \$40,000 - \$49,999
 E = \$50,000 - \$74,999
 F = \$75,000 - \$99,999
 G = \$100,000 - \$124,999
 H = \$125,000 or more

Married - Marital Status

M=Married
 S=Single
 A=Inferred Married
 B=Inferred Single

WealthScr - Model based on income, homeownership, and other assets owned [A - H]

A = Estimated Net less than \$5,000
 B = Estimated Net \$5,000 - \$19,999
 C = Estimated Net \$20,000 - \$49,999
 D = Estimated Net \$50,000 - \$79,999
 E = Estimated Net \$80,000 - \$99,999
 F = Estimated Net \$100,000 - \$249,999
 G = Estimated Net \$250,000 - \$499,999
 H = Estimated Net over \$500,000

DwellType - Dwelling Type [1 char]

S=Single Family Dwelling Unit (SFDU)
 M=Multi-family Dwelling Unit (MFDU)

MrktHomeVal - Estimated Home Market value [1 char]

A = \$1,000 - \$24,999
 B = \$25,000 - \$49,999
 C = \$50,000 - \$74,999
 D = \$75,000 - \$99,999
 E = \$100,000 - \$124,999
 F = \$125,000 - \$149,999
 G = \$150,000 - \$174,999
 H = \$175,000 - \$199,999
 I = \$200,000 - \$224,999
 J = \$225,000 - \$249,999
 K = \$250,000 - \$274,999
 L = \$275,000 - \$299,999
 M = \$300,000 - \$349,999
 N = \$350,000 - \$399,999
 O = \$400,000 - \$449,999
 P = \$450,000 - \$499,999
 Q = \$500,000 - \$749,999
 R = \$750,000 - \$999,999
 S = \$1,000,000+

ChildCd - Presence of Children [Y or blank].

Y= children present.

ChildNbrCd - Number of Children [1 char]

A=No children
 B=less than 3
 C= 3-5 children

MHV - Median House Value Code [1 char]

A = Less than \$50,000
 B = \$50,000 - \$99,999
 C = \$100,000 - \$149,999
 D = \$150,000 - \$249,999
 E = \$250,000 - \$349,999
 F = \$350,000 - \$499,999

G = \$500,000 - \$749,999

H = \$750,000 - \$999,999

I = \$1,000,000+

MedSchl - Median years of school [3 digits max, 0 - 999]

PurPowerScr - Purchasing Power Model, 1 character:

A = (820-High)

B = (800-819)

C = (780-799)

D = (760-779)

E = (740-759)

F = (720-739)

G = (700-719)

H = (680-699)

I = (660-679)

J = (640-659)

K = (620-639)

L = (600-619)

M = (580-599)

N = (560-579)

O = (540-559)

P = (520-539)

Q = (500-519)

R = (480-499)

S = (460-479)

T = (Low-459)

API ERROR MESSAGES

ERROR MESSAGE	MEANING
D247_INVALID_CREDENTIALS	Wrong key value provided
D247_INVALID_API_KEY	
D247_INVALID_SRC_IP	IP Authentication is turned on for your account, but the source IP address is not in the list of allowed addresses.
D247_ACCOUNT_INACTIVE	Account set to inactive, possible reasons could be expired subscription, no funds left on the account
D247_INSUFFICIENT_FUNDS	Not enough funds left on the account.
D247_MISSING_REQUIRED_PARAMS	Not all required parameters for specific api is provided or provided parameters are invalid
D247_PROCESSING_ERROR	Some unknown error occurred while processing your request
D247_INVALID_API	Wrong api parameter value provided
D247_NO_DATA_FOUND	No related data found on our database
D247_MISSING_PHONE_NUM	Data service you requested requires a phone number, but you didn't provide it
D247_INVALID_PHONE	The phone number provided is not valid
D247_INVALID_EMAIL	The email address provided is not valid
D247_CANNOT_ADD_NUMBER	The phone number you provided cannot be added to DNC list
D247_CANNOT_REMOVE_NUMBER	The phone number you provided cannot be removed from a DNC list
D247_INVALID_ZIPCODE	The zip code specified does not exist
D247_NO_SUBCLIENT_DATA	
D247_INVALID_IP_ADDRESS	
D247_ADDRESS_NOT_FOUND	
D247_INVALID_ADDRESS	

D247_DATA_RESTRICTED	
D247_NOT_IMPLEMENTED	
D247_NO_CANADA_ACCESS	
D247_TOO_MANY_QUERIES	

EXAMPLES IN VARIOUS PROGRAMMING LANGUAGES

cURL

single query:

```
curl 'https://api.data247.com/v3.0?key=MyAPIKey&api=CU&phone=16175551212'
```

multiple queries in a single API call:

```
curl -i -X POST -H "Content-Type:application/json" \
-d '{
  "CU": [
    {
      "phone": "15084041234"
    },
    {
      "phone": "17818640001"
    }
  ]
}' 'https://api.data247.com/v3.0?key=MyAPIKey&api=CU'
```

Perl

```
use REST::Client;

my $client = REST::Client->new();

$content = {'Content-type' => 'application/json'};
$url="http://api.data247.com/v3.0";
$req = '{
  "key" : "",
  "api" : "MT",
  "phone": "17818640108"
}';
$client->POST($url,$req,$content);
print $client->responseContent();
```

PHP

```
<?php

$key = "MyAPIKey"; // Replace key value with your own api key
$phone = "17818641234";

$url = "https://api.data247.com/v3.0?key=$key&api=MT&p1=$phone";
$result = @file_get_contents($url);
```

```

if ($result){
    $result = @json_decode($result, true);
    if (!empty($result['response']['status']) && $result['response']['status'] == 'OK'){
        $response = $result['response']['results'][0];
        echo "Phone    : $phone\n";
        echo "carrier  : " . $response['carrier_name'] . "\n";
        echo "Is wireless: " . $response['wless'] . "\n";
        echo "SMS gateway: " . $response['sms_address'] . "\n";
    } else {
        echo "Error: " . $result['response']['status'] . "\n";
    }
} else {
    echo "Error calling API\n";
}
?>

```

PYTHON

```

import requests

key = "MyAPIKey"
phone = "7818641234"

try:
    r = requests.get("https://api.data247.com/v3.0?key=%s&api=MT&p1=%s" % (key,phone))
    adict = r.json()['response']
    if 'status' in adict and adict['status'] == 'OK':
        result = adict['results'][0]
        print("Phone number: %s" % (phone))
        print("Carrier Name: %s" % (result['carrier_name']))
        print("Is wireless : %s" % (result['wless']))
        print("SMS gateway : %s" % (result['sms_address']))
    else:
        print("Error: %s \n" % (adict['status']))
except Exception as err:
    print("Error: "+ str(err))

```

JAVA

```

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class test {
    public static void main(String[] args) {
        try {

            String apikey = "";

            URL url = new URL("https://api.data247.com/v3.0?key=" + apikey + "&api=MT&phone=17815551212");
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setRequestMethod("GET");

```

```
conn.setRequestProperty("Accept", "application/json");
if (conn.getResponseCode() != 200) {
    throw new RuntimeException("Failed : HTTP Error code : " + conn.getResponseCode());
}
InputStreamReader in = new InputStreamReader(conn.getInputStream());
BufferedReader br = new BufferedReader(in);
String output = br.readLine();
System.out.println("Output = " + output);
conn.disconnect();

} catch (Exception e) {
    System.out.println("Exception in example:- " + e);
}
}
```

MIGRATING FROM DATA247 API VERSION 2.0 to 3.0

Coming soon